

---

## ■ ■ PROJECT RAG (6 PROGRESS)

---

### ■ PROJECT PROGRESS 1 — Setup Infrastruktur Dasar

#### Instruksi

Pada progress ini Anda wajib menyiapkan seluruh infrastruktur dasar untuk proyek RAG. Tugas meliputi:

1. Instalasi tools dasar:
  - Docker & Docker Compose
  - Node.js
  - n8n (local)
  - VSCode
  - Git + GitHub
  - **Ngrok** (tambahan baru)
2. Setup repository GitHub untuk project RAG
3. Struktur folder awal proyek
4. Dokumentasi instalasi dan arsitektur awal

#### Yang Dikumpulkan

- Link GitHub repository
- Screenshot instalasi tools
- Diagram arsitektur dasar
- Laporan setup (1 halaman)

---

### ■ PROJECT PROGRESS 2 — Workflow 1 (Telegram / Webhook → n8n → OpenAI)

#### Instruksi

Pada progress ini Anda membuat **Workflow 1** dengan menggunakan **Telegram Bot** dan **OpenAI** melalui n8n:

1. Akses n8n melalui URL **Ngrok**

2. Buat workflow:
  - Input dari Telegram Bot
  - Proses pertanyaan via OpenAI
  - Output kembali ke Telegram
3. Uji workflow menggunakan Telegram
4. Push workflow ke GitHub

### **Yang Dikumpulkan**

- Link GitHub (berisi file workflow + dokumentasi)
  - Screenshot hasil percobaan Telegram
  - URL Ngrok
  - Laporan singkat workflow
- 

## **PROJECT PROGRESS 3 — Workflow 2 (Local WebApp → Webhook → n8n → OpenAI)**

### **Instruksi**

Pada progress ini Anda membangun **Workflow 2** untuk WebApp:

1. Buat WebApp sederhana di lokal (Node.js / Next.js / Python Flask)
2. WebApp mengirim pertanyaan ke endpoint Webhook n8n
3. n8n memproses via OpenAI (tanpa Pinecone)
4. Push WebApp ke GitHub
5. **Persiapkan WebApp untuk deployment ke Vercel**

### **Yang Dikumpulkan**

- Link GitHub WebApp
  - Link GitHub workflow
  - Screenshot WebApp lokal memanggil n8n
  - Penjelasan singkat alur data
- 

## **PROJECT PROGRESS 4 — Workflow 3 (Google Drive → Embedding → Pinecone)**

## Instruksi

Progres ini fokus membuat **database vector**:

1. Workflow untuk membaca file dari Google Drive
2. Generate embedding menggunakan OpenAI
3. Simpan vektor ke Pinecone
4. Push Workflow 3 ke GitHub
5. Dokumentasikan skema data dan arsitektur

## Yang Dikumpulkan

- Link GitHub workflow
  - Screenshot hasil embedding
  - Screenshot data masuk ke Pinecone
  - Laporan singkat
- 

## PROJECT PROGRESS 5 — Integrasi Workflow 1 + Workflow 2 + Workflow 3

### Instruksi

Pada progress ini Anda melakukan integrasi total:

- Telegram → Workflow 1 → Pinecone
- WebApp → Workflow 2 → Pinecone
- Cek alur end-to-end

### Yang Dikumpulkan

- Screenshot Telegram berhasil mengambil jawaban berbasis Pinecone
  - Screenshot WebApp berhasil memanggil Pinecone
  - Link GitHub update
  - Laporan integrasi (maks 1 halaman)
- 

## PROJECT PROGRESS 6 — Integrasi Cloudflare (DNS + Tunnel + Security)

### Instruksi

Pada progress ini Anda mengganti Ngrok dengan **Cloudflare** serta menambahkan lapisan keamanan jaringan.

### **Tugas Detail:**

#### **1. Setup Cloudflare Free Plan**

- Tambahkan domain
- Atur DNS
- Aktifkan proxy (oranye)

#### **2. Setup Cloudflare Tunnel**

- Install cloudflared
- Buat tunnel
- Routing domain:
  - `n8n.<domain>.com` → `localhost:5678`
  - `webhook.<domain>.com` → `endpoint WebApp/n8n`

#### **3. Integrasi ke Workflow 1 & 2**

- Update Telegram webhook → URL Cloudflare
- Update WebApp Vercel → URL Cloudflare
- Hentikan pemakaian Ngrok

#### **4. Firewall Rules (minimal 2 rules)**

Contoh:

- Blokir method HTTP selain GET/POST
- Izinkan request hanya dari Vercel dan Telegram

#### **5. Monitoring Cloudflare Analytics**

- Traffic
- Firewall events
- Latency

#### **6. Push ke GitHub**

Folder baru: `/cloudflare/`

Berisi:

- `tunnel-config.yaml`

- firewall-rules.md
- Dokumentasi arsitektur baru

### **Yang Dikumpulkan**

- Link GitHub repo (wajib updated)
  - Screenshot DNS, Tunnel, Firewall rules, Analytics
  - URL n8n via Cloudflare
  - URL WebApp via Cloudflare
  - Laporan integrasi Cloudflare (1 halaman)
- 

### **📁 Format Pengumpulan**

Setiap progress hanya dikumpulkan pada **kanal Project yang sesuai**, contoh:

- **Project Progress 1**
- **Project Progress 2**
- ...
- **Project Progress 6**

Semua berbentuk:

- PDF laporan
  - Link GitHub
  - Screenshot bukti
  - Link sistem yang berjalan
-